CHAPTER

2

THE STRUCTURE OF ACTIVE DIRECTORY

After completing this chapter, you will be able to:

- Understand the purpose and role of sites in Active Directory
- Understand the purpose and role of Organizational Units within Active Directory
- Understand the purpose and role of containers within Active Directory
- Understand the purpose and role of domain trees and forests within Active Directory
- Understand the purpose and role of DNS and DDNS within Windows 2000
- Understand the logical structure of Active Directory
- Understand the physical structure of Active Directory

This chapter will discuss the major components of Active Directory, including sites, containers, Organizational Units, domain trees and forests, bridgehead servers, and more. The purpose of each of these components and their roles within the Active Directory structure will be covered. In addition, we will discuss both the logical and physical structure of Active Directory and the implications of these structures on the domain design.

Active Directory Terminology

With the advent of Windows 2000, everything about Windows domains has changed. Rather than the simple domain structures of Windows NT, Windows 2000 supports the Active Directory service. The increased complexity of Active Directory naturally requires a richer vocabulary. In addition to the domains and trusts familiar to Windows NT administrators, new terms such as *sites*, *schema*, *forests,* and *containers* are included within an Active Directory structure. We'll discuss these new terms, precisely how each is used, and the implication of these new structures to the traditional Windows NT network environment. First, let us begin by defining some terms.

Domains

A **domain** is a selection of computers, user accounts, or other objects that share a common security boundary. This means that every item within a domain is controlled by the same security policies and access restrictions. The concept of a domain as the core of the networking structure was introduced in Windows NT. A Windows NT domain is a self-contained unit, and security policies do not extend beyond the borders of the domain. However, Windows 2000 has extended the concept of a domain somewhat beyond that of Windows NT.

The components of a Windows 2000 domain are as follows:

- A hierarchical structure of containers and objects (we'll define these terms shortly).
- A unique Domain Name System (DNS) domain name. As we will discuss in Chapter 4, the NetBIOS name resolution and naming conventions used in earlier Microsoft operating systems are no longer required in a native Windows 2000 environment. Instead, Windows 2000 uses DNS names exclusively for name resolution, and the DNS domain name is used as the Windows domain name.
- A security boundary that controls authentication of users, access to resources, and any trusts with surrounding domains.

A Windows 2000 domain includes every object within a domain. An **object** is anything within the network that has been defined and can be uniquely identified within the Windows 2000 hierarchy. Although this may seem to be a recursive definition, it really is not. An object can represent a wide range of items: Computers, network users, printers, shared folders, and individual files are all examples of objects. A domain is simply a collection of these objects, all of which are controlled via a central security policy. The administrator of the domain defines the security policy for the domain and the access rights relating to the objects in the domain. Each domain has its own security policy; access to resources across domains is accomplished via security relationships called **trusts**.

Domain Controllers

The rights of a particular user or group of users within a domain are determined by the security policy within that domain. Likewise, resources and the access to those resources are defined by that security policy. When a user logs in to a domain, he or she is authenticated and assigned the appropriate rights. Servers known as **domain controllers** (DCs) provide this authentication.

DCs are servers that are running Windows 2000 and on which the Active Directory service is installed. Each DC possesses a copy of the domain's security policy, lists of domain users and passwords, and the access requirements for each object within the domain. The information about the domain is stored in a database called the **data store** on each DC. The data store is contained within the file ntds.dit. The default location for this file is %*systemroot*%\NTDS, but it can be relocated during the installation of the Active Directory service.

In a homogenous Windows 2000 domain, each of the DCs acts as an equal peer and can send updated domain objects or rights to other DCs. DCs can also interoperate with earlier versions of Windows NT, such as NT 4 and NT 3.51. In these mixed-mode environments, the earlier versions of Windows NT operate as backup DCs and act as passive receivers of updates from a Windows 2000 DC. One specific Windows 2000 DC takes on the role of the Primary Domain Controller (PDC) for the downlevel operating systems (Windows NT, Windows 9*x*, and Windows 3.*x*). This *PDC emulation master* is assigned to act as the Windows NT PDC and replicates directory changes to Windows NT DCs. In addition, the PDC emulation master provides network services for network clients that cannot access Active Directory. These clients include Windows NT Server and Workstation, Windows 98 and 95, and Windows 3.*x*/DOS operating systems.

Trust Relationships

Windows NT domains are independent entities. The security policies and rights within a domain are limited to the resources within the domain. Although this structure made sense during NT's early role as a workgroup and small local area network (LAN) server, the same structure became a limiting factor as Windows NT evolved into an enterprise network operating system (NOS). Massive network environments strain the capacities of the Windows NT security database and can require multiple domains to contain all the computer accounts and user accounts.

Because the security policies within a domain are limited to that domain only, Windows NT administrators are forced to use relationships called *trusts* to enable cross-domain access to resources. Trusts allow network administrators to assign rights and resources to users that are authenticated within another domain. A trust relationship consists of two domains: a trusted domain that authenticates a user, and a trusting domain that accepts users authenticated by another domain. As shown in Figure 2-1, Domain 2 trusts the authentication of Domain 1 and will allow Domain 1 users to access resources within Domain 2.

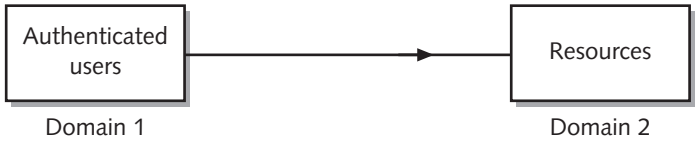

Figure 2-1 One-way trust relationship

The actual access rights within Domain 2 are still controlled by the security policy enacted by the administrator of Domain 2.

In a traditional Windows NT trust scenario, the trusts can be either one way or two way. A one-way trust works as just described: One domain accepts (trusts) the authenticated users from a second (trusted) domain. If both domains accept each other's authenticated users, the relationship is described as a two-way trust relationship. An example of a two-way trust is shown in Figure 2-2.

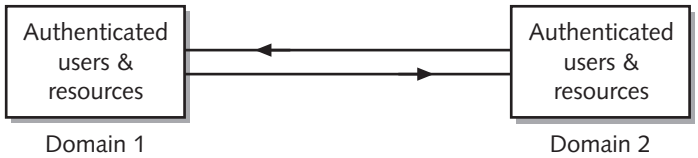

Figure 2-2 Two-way trust relationship

Trust relationships are effective in providing cross-domain resource access, but limitations with the implementation of trust relationships within NT 4 and earlier resulted in excessive administrative overhead in larger environments. The main problem with Windows NT trusts is the lack of transitive trusts. If Domain 1 trusts Domain 2, and Domain 2 trusts Domain 3, no trust relationship exists between Domain 1 and Domain 3. If Domain 3 users need access to Domain 1 resources, a separate trust relationship must be established between the domains.

As you might suspect, the trust relationships within a large network environment can easily become very complicated. Furthermore, each trust relationship requires administrative intervention at both ends to form the relationship. In some environments, adding a new domain requires that trusts be configured with every other domain within the network. Windows NT administrators developed several methods to improve the efficiency of the trusts, including the master/resource domain model and the multimaster model. Ultimately, however, these methods are merely elegant workarounds for an inherent problem.

Windows 2000 extends the capability of trusts by allowing transitive trusts. With a transitive trust, if Domain 1 trusts Domain 2, and Domain 2 trusts Domain 3, then a trust

relationship automatically exists between Domain 1 and Domain 3. This relationship is illustrated in Figure 2-3. These transitive trusts eliminate much of the web of trusts required within a Windows NT environment. Furthermore, the two-way transitive trusts are created automatically when a new domain is added to a domain tree. We will discuss domain trees and the roles of trusts within those trees shortly.

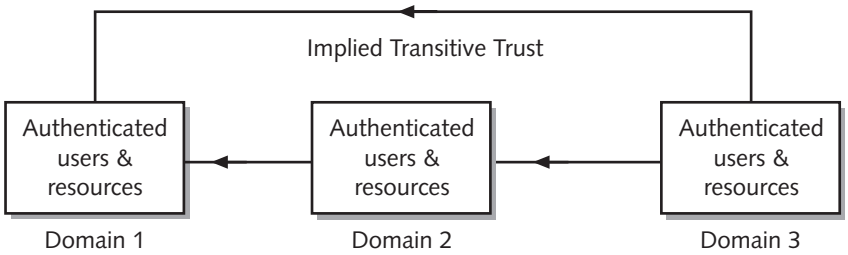

Figure 2-3 Transitive trust relationship

Namespace

Windows 2000 uses a different method of name resolution and domain organization than earlier versions of Windows NT. Rather than the NetBIOS-based name resolutions, Windows 2000 uses DNS name resolution. The DNS system is the same as that used for resolving Internet domain names to Internet Protocol (IP) addresses; most people will recognize www.microsoft.com as an example of a DNS-resolvable name.

DNS is a hierarchical naming system, with the names becoming more specific to the left and more general towards the right. Let us examine a DNS name in more detail first, and then look at the implications for the Windows 2000 domain structure.

We'll use a hypothetical company for our study. Texas Pinball and Cattle Company sells and services pinball machines to homes and businesses. As do most businesses these days, the company has a Web site and e-mail connectivity. Traditionally, the company has used a Windows NT domain named Corporate for its internal network, and its main file servers are known as Flipper and Bumper. See Figure 2-4 for the structure of the network.

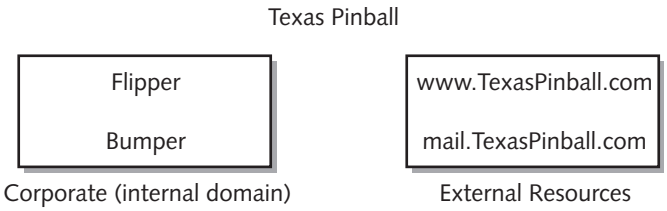

Figure 2-4 Texas Pinball (traditional NT structure)

Texas Pinball has two Internet-accessible resources: the Web site at www.TexasPinball.com and the mail server at mail.TexasPinball.com. The DNS system translates these names into IP addresses that remote computers use to connect to the servers. Taking the names from right to left, the .com represents a top-level domain (TLD). Top-level domains are used to distinguish the type of organization represented by the domain. The traditional uses of TLDs are listed in Table 2-1. Over the past few years, the traditional demarcation between the .com, .net, and .org addresses has faded as the number of .com addresses has increased. Many organizations now register .net and .org names for a variety of reasons, such as protecting their .com address from similar names, a lack of desirable names in the .com namespace, and duplicate organization names.

Table 2-1 Top-level domain uses

Type	Meaning
.com	Commercial entities
.net	Internet infrastructure services
.org	Nonprofit organizations
.edu	Universities, colleges, and other educational institutions
.mil	United States military
.gov	United States government

In addition to the TLDs listed in Table 2-1, each country has been assigned a country code that also functions as a TLD. Although use of a country code is rare within the United States, organizations within other countries use country codes regularly. An example of this is www.microsoft.co.uk.

The next section of the domain name is TexasPinball. This portion is commonly known as the second-level domain name, and it is indicative of the company or organization. Everything to the left of the second-level domain name is under the control of the organization that owns that domain name, including servers and subdomains. If the name to the left of the domain name is a server, then it is referred to as a **hostname**. If the name to the left of the domain name refers to another collection of computers, it is considered a **subdomain**. Figure 2-5 shows the relationship between the levels of a domain name.

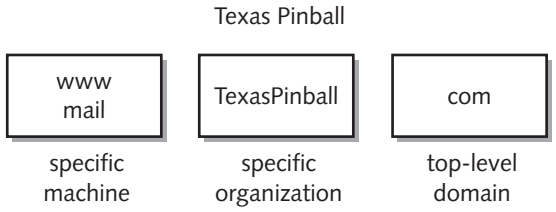

Figure 2-5 A diagram of DNS space

As we mentioned earlier, a company or organization controls the naming within its second-level domain name. Every host and subdomain within the domain name is considered part of the domain's **namespace**.

To bring this discussion back to our example, Texas Pinball and Cattle Company controls the namespace that is associated with the TexasPinball.com domain. With the traditional NT domain structure and the NetBIOS naming associated with it, no direct relationship exists between NT domain names and DNS domain names. Therefore, Texas Pinball and Cattle Company has only two objects within its namespace: www.TexasPinball.com and mail.TexasPinball.com. The servers and workstations within the Corporate domain are not considered part of the TexasPinball.com namespace.

With Windows 2000, DNS is now the primary method of name resolution; as a result, Windows domains now map much closer to DNS domains. Let's take a look at how our example company might look with Windows 2000.

Texas Pinball and Cattle Company can use its existing DNS domain name for internal use, or the company can choose to use a separate DNS domain for its internal network. For now, we will assume that the company will use its existing TexasPinball.com domain for both the internal and external network.

Remember that both subdomains and hosts are part of the namespace controlled by a company. As a result, it is possible to design a domain structure that incorporates both the internal network and the externally accessible resources within the same DNS domain. In the example shown in Figure 2-6, the internal network is designated by the Corporate.TexasPinball.com subdomain. The two fileservers mentioned earlier are named Flipper.Corporate.TexasPinball.com and Bumper.Corporate.TexasPinball.com.

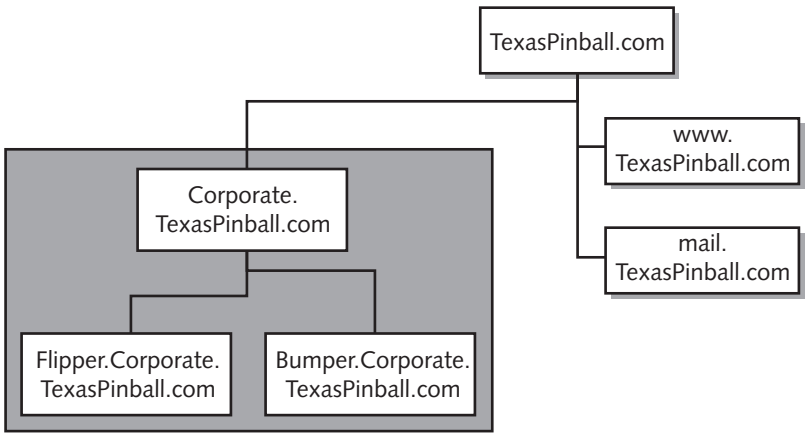

Figure 2-6 Windows 2000 namespace

Dynamic DNS

With the move from NetBIOS naming to DNS naming, a significant change has occurred in the way clients interact with the name servers. With previous versions of Windows NT, client operating systems register themselves with a Windows Internet Naming Service (WINS) server that maintains a database of NetBIOS names and IP addresses. WINS clients automatically reregister themselves whenever the IP address changes.

On the other hand, computers running the Windows 2000 operating system can register themselves with a server running the DNS service. The ability of a client to register itself into a DNS host table as it joins the network environment is a hallmark of the Dynamic DNS (DDNS) system. In addition, if the IP address of a computer changes, the computer can automatically register the change with the DNS server.

In a homogenous Windows 2000 environment, DDNS replaces the WINS services required in earlier Windows NT environments. If the network includes downlevel operating systems such as Windows NT, Windows 95 or 98, or Windows 3.*x*, a WINS server is still required for name resolution.

Domain Trees

The use of DNS namespace and the Active Directory structure require us to rethink the traditional use and structures of Windows domains. Remember that Windows 2000 will automatically form two-way transitive trusts between domains if they have the same DNS root (i.e., Corporate.TexasPinball.com and Sales.TexasPinball.com). Remember also that the use of DNS for naming within Windows 2000 allows the use of hierarchical domain names, such as Home.Sales.TexasPinball.com.

The combination of these two elements creates domain trees. A **domain tree** is a group of Windows 2000 domains that share the same namespace, a common schema, and a Global Catalog. (We will discuss the schema and Global Catalog later in this chapter.) In addition, the domains within a domain tree will usually share an automatic transitive trust with other domains within the tree. The result is that permissions and rights flow down throughout the tree and allow for administrative control throughout the domain. Figure 2-7 shows the relationships and structure within a domain tree.

The elements of a domain tree are as follows:

- The domains share a common namespace. Usually, the domain tree consists of parent and child domains.
- All domains share a common schema.
- All domains share a common Global Catalog.
- Implicit two-way transitive trusts exist between domains.

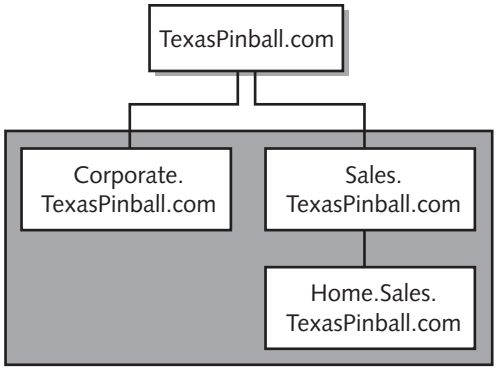

Figure 2-7 Windows 2000 domain tree

Domain Forests

Just as a normal forest is a collection of trees, so too is a **domain forest** a collection of domain trees. Recall that a domain tree is a collection of domains that share a common schema, Global Catalog, and namespace. However, if the domains do not share a common namespace, but they still share a common schema and Global Catalog, they are considered part of a domain forest. All domains within a domain forest share implicit two-way transitive trusts with the other domains within the forest.

The elements of a domain forest are as follows:

- The domains have a noncontiguous namespace and differing name structure.
- All domains share a common schema.
- All domains share a common Global Catalog.
- Domains operate independently, but cross-domain communication is enabled by the forest.
- Implicit two-way transitive trusts exist between domains and domain trees.

Domain forests often arise in situations where companies have different internal and external domain names, or in cases where mergers, acquisitions, or other elements have resulted in multiple root domains within an organization. As an example, let us imagine that Texas Pinball and Cattle Company decides to acquire an Internet-based pinball service dispatching company. The pinball service company is called Pinball.nu ("We make your Pinball like NU"), and naturally it uses the Pinball.nu namespace. Rather than force a migration to the TexasPinball.com namespace, the network administrators instead choose to form a Windows 2000 domain tree that includes both the TexasPinball.com domain and the Pinball.nu domain, as shown in Figure 2-8. Once the domains are associated within a tree, the resources of all the domains are accessible by the users of the two organizations.

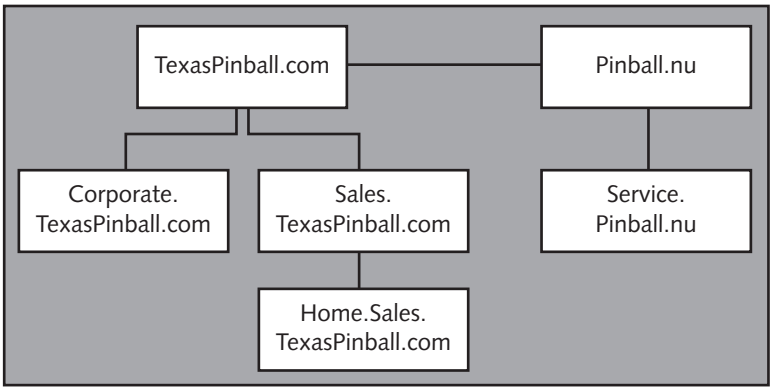

Figure 2-8 Domain forest

Active Directory Components

The Active Directory service allows administrators to create a network structure that matches the needs of the organization. You do so by defining objects and their related attributes within a directory structure. The directory and all its components are replicated to all the DCs within the domain. Each DC stores a copy of the directory and the security policy of the domain, and metes out access as defined by the security policies.

Active Directory Objects

Within Active Directory parlance, an **object** is simply a defined element within the directory. Each object refers to a specific, distinctive, named network resource. Each resource is defined by a set of attributes that are characteristic for that resource, such as computer names or user passwords.

Almost everything within Active Directory is considered an object. User accounts, computer accounts, printers, shares, servers, containers, and the like can all be viewed as objects within the Active Directory tree. Many types of objects are predefined, such as user accounts and computer accounts.

Objects can be collected and organized within the directory. Logical groupings of similar objects are considered **classes**. Some objects can contain other objects, and are naturally known as **containers**. One example of a container object is the domain, which contains user accounts, servers, computers, and other objects.

Active Directory Schema

Recall from when we were discussing domain trees and forests that one of the defining elements of a forest or tree was a common schema. The **schema** is simply a definition of the types of objects allowed within a directory and the attributes associated with those

objects. As you can see, these definitions must be consistent across domains in order for the security policies and access rights to function correctly.

Two types of definitions exist within the schema: **attributes** and **classes**, also known as **schema objects** and **metadata**. Attributes are defined only once, and then can be applied to multiple classes as needed. The object classes, or metadata, describe which attributes are used to define objects. As an example, the Users class requires certain attributes such as user name, password, groups, and so on. A particular user account is simply an object that has those attributes defined.

As shown in Figure 2-9, a class is simply a generic framework for particular objects. That framework is generated by a collection of attributes, such as Logon Name and Home Directory for users, or Description and Network Address for computers. Windows 2000 ships with a predefined set of attributes and classes that fit the needs of many network environments. In addition, you can expand the schema by defining additional attributes and extending the classes within the directory.

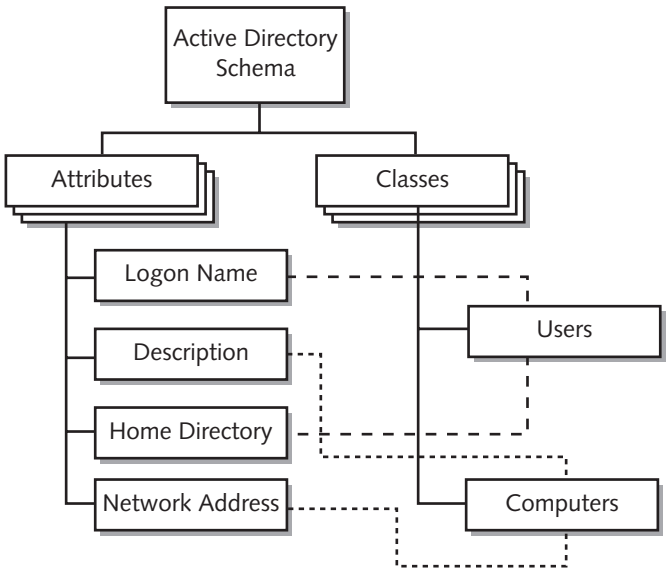

Figure 2-9 The relationship between attributes and classes within a schema

Extending a schema by developing additional attributes and classes is an advanced function that should be performed only after careful planning. Remember that the changes are automatically replicated within the directory, and that a schema cannot be deleted.

Organizational Unit

One of the enhancements within Active Directory is the ability to organize your network in a logical manner and hide the physical structure of the network from the end users. Active Directory uses a special container known as an **Organizational Unit** (OU) to organize objects within a domain in administrative groups. These OUs can be used to divide a domain into groups that mirror the functional or physical separations within the company. OUs are limited to a single domain; each domain can implement its own OU hierarchy, as illustrated in Figure 2-10.

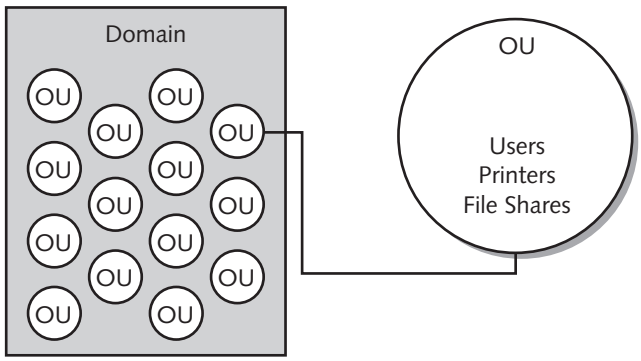

Figure 2-10 Organizational Units within a domain

An OU can contain user accounts, computers, printers, file shares, applications, and any other objects within the domain. OUs can be used to separate administrative functions within a domain without granting administrative rights to the whole domain.

An OU is the smallest element to which you can assign administrative rights. Therefore, OUs can be used to delegate authority and control within a domain; in essence, OUs allow the functionality of subdomains without your actually having to create additional domains.

Global Catalog

Domain controllers keep a complete copy of the Active Directory database for a domain, so that information about each object in the domain is readily available to the users and services. This arrangement works well for the local domain, but what about information about other domains and the objects within those other domains? Remember, one of the design goals for Windows 2000 was a unified logon, no matter where a user was located within the domain tree. Obviously, for such a unified logon to work, the local DCs must have some information about the other domains within the tree and forest. However, replicating all the information about all the objects in all the domains within a forest is simply not feasible.

Windows 2000 solves this issue through the use of a special limited database known as the **Global Catalog**. The Global Catalog stores partial replicas of the directories of

other domains. The catalog is stored on DCs that have been designated as Global Catalog Servers. These servers also maintain the normal database for their domain.

Function of the Global Catalog

The Global Catalog has two primary functions within Active Directory. These functions relate to the logon capability and the queries within Active Directory. We will next examine each function in detail.

Within a native-mode multidomain environment, the Global Catalog is required for logging on to the network. The Global Catalog provides universal group membership information for the account that is attempting to log on to the network. If the Global Catalog is not available during the logon attempt, and the user account is external to the local domain, the user will only be allowed to log on to the local machine.

Obviously, if the account is part of the local domain, the DCs for the local domain will handle the authentication request. The Global Catalog is required only when a user account or object needs to be authenticated by another domain.

Queries make up the majority of Active Directory traffic, and queries for objects (such as printers and services) occur much more often than database updates. Within a simple single-domain environment, the directory is readily available for these queries. However, imagine for a moment a highly complex multidomain environment. It doesn't make sense to require every query to search through each domain.

The Global Catalog maintains a subset of the directory information available within every domain in the forest. This arrangement allows queries to be handled by the nearest Global Catalog, and thus saves time and bandwidth. If more than one DC is a Global Catalog Server, the response time for the queries improves. Unfortunately, each additional Global Catalog Server increases the amount of replication overhead within the network.

The Global Catalog is a read-only database, unlike the normal directory database.

Global Catalog Servers

Windows 2000 automatically creates a Global Catalog on the first domain controller within a forest. Each forest does require at least one Global Catalog. In an environment with multiple sites, it is good practice to designate a DC in each site to function as a Global Catalog Server. Remember, native-mode Windows 2000 domains require a Global Catalog to allow users to complete the authentication process and log in to the network. A mixed-mode domain does not require a Global Catalog server.

If you need additional Global Catalog Servers, you can add the service to any DC within the forest. You do so through the AD Sites and Services snap-in. The Global Catalog can also be moved off the initial DC if additional DCs are available.

Although any and all DCs can be configured as Global Catalog Servers, a sense of balance is necessary when designating these servers. As the number of Global Catalog Servers increases, the response time to user inquiries decreases. However, the replication requirements within the environment increase as the number of Global Catalog Servers increases.

Operation Masters

Much of the replication within an Active Directory environment is multi-master replication, which means that the DCs are all peers. This is in contrast to earlier versions of Windows NT, in which a Master DC is responsible for recording all changes to the security policy and replicating those changes to the backup DCs.

Several types of operations are impractical for a multi-master environment, however. Windows 2000 handles these operations by allowing only a single DC to make these types of changes. This DC is known as an **operations master**. Actually, five different operation master roles can be assigned to DCs: schema master, domain naming master, relative ID master, PDC emulator, and infrastructure master. Each of these roles will be discussed in detail later in this chapter.

By default, Windows 2000 assigns all five of these operations master roles to the first DC installed in a forest. In a small network environment, these roles may very well stay with that first DC. As the network environment gets larger, some of the roles will need to be reassigned to other DCs. Two of the operations master roles must appear in each domain forest, and the remaining three must appear in each domain.

The two operations master roles assigned on a forestwide basis are the schema master and the domain naming master. Only one of each of these operation masters can exist within a forest. The schema master controls all the updates and modifications to the schema itself. As you will recall, the schema controls the definition of each object in the directory and its associated attributes. The domain naming master controls the addition of domains to or removal of domains from the forest.

The three operations master roles assigned for each domain are the relative ID master, the PDC emulator, and the infrastructure master. Each domain within a forest must have one of these masters.

The relative ID (RID) master controls the sequence number for the DCs within the domain. The master assigns a unique sequence of RIDs to each of the DCs. When a new object is created by a DC, the object is assigned a security ID (SID). The SID must be unique within the domain. The SID is generated by combining a domain security ID and a relative ID. The domain security ID is a constant ID within the domain, whereas the relative ID is assigned to the object by the DC. When the DC uses all the RIDs that the RID master has assigned, the DC receives another sequence of RIDs from the RID master.

The PDC emulator is used whenever a domain contains non–Windows 2000 computers. It acts as a Windows NT PDC for downlevel clients and for Windows NT backup DCs. The PDC emulator processes password changes and receives preferential treatment

within the domain for password updates. This preferential treatment continues even if the domain is operating in native mode. If another DC is unable to authenticate a user due to a bad password, the request is forwarded to the PDC emulator.

The infrastructure master is responsible for maintaining all interdomain object references. It informs certain objects (such as groups) that other objects (such as users in another domain) have been moved, changed, or otherwise modified. This update is needed only in a multiple domain environment. If there is only a single domain, then all DCs already know of the update, and this role is unnecessary. Likewise, if all DCs are also Global Catalog servers, the DCs are aware of the updates and do not need the assistance of the infrastructure master.

Physical Structure of Active Directory

The primary goal of the Active Directory service is to allow users to access resources throughout the domain structure without necessarily knowing precisely where those resources are located. Active Directory hides the physical structure of the network from the end user and shows instead a logical topology that focuses more on the resources than their location. Within this section, however, we will focus on the physical structure of a Windows 2000 network and its related elements: sites, bridgehead servers, DCs, and site links.

Sites

The core concept of the physical structure of a Windows 2000 domain is the site. A **site** is a collection of computers connected via a high-speed network. Typically, the computers within a site are connected via LAN-style technology, and are considered to be well-connected. **Well-connected** generally means constant high-speed connectivity within an IP subnet, although a site can include multiple subnets.

It is important to understand that sites and domains do not have a direct relationship. Domains map the logical structure of your organization, whereas sites relate to the physical layout of the network. The domain namespace is likewise unrelated to the physical sites, although many times administrators will choose to align the namespace and the physical sites during the planning phase of a Windows 2000 rollout or migration.

A site can contain multiple domains; likewise, a domain can cross several sites. In most cases, sites will mirror the actual physical layout of the network, with a site at each of the major business locations of a company. If a company is using a single-domain structure, then that domain will cross the sites as shown in Figure 2-11.

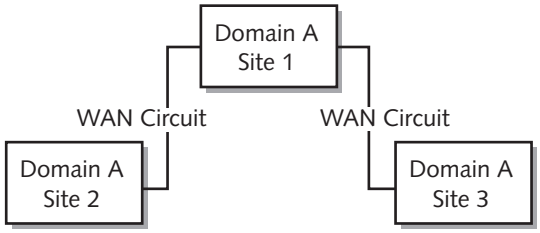

Figure 2-11 A single domain across multiple sites

Because of the separation of the physical and logical structure, a site can also support multiple domains. In Figure 2-12, Site 1 has computers in both Domain A and Domain B.

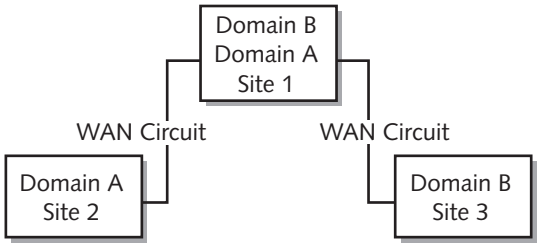

Figure 2-12 Multiple domains across multiple sites

Why Use Sites?

If you are a network administrator who has supported wide area network (WAN)-connected Windows NT domains, you may wonder about the purpose of defined sites within Windows 2000. After all, the use of remote networks, backup DCs, and WAN circuits certainly is nothing new, nor is the concept of multiple domains at a particular location. The problem with the older Windows NT network lies in the replication of security information between the DCs. Whenever changes occur to the security policy within a domain—such as new user accounts, new groups, or even group membership changes—the entire Security Accounts Manager (SAM) database must be replicated across the WAN link. In large or active network environments, this replication can consume a majority of the bandwidth between locations.

Windows 2000 corrects this issue by replicating data between DCs differently depending on the relationship between the DCs. Within a site, the primary goal of replication is to keep the DCs updated with as little latency as possible. Between sites, the replicated data is compressed and sent periodically. The compression helps save bandwidth, but does require more processing overhead on the part of the DCs.

The primary function of a site is to consolidate directory service requests within a high-speed connection area and to control replication with external DCs. Sites provide the following benefits:

- Directory services are provided by the closest DC, if one is located within the site.
- Latency is minimized for replication within a site.
- Bandwidth utilization for replication is minimized between sites.
- Replication can be scheduled between sites to better suit network utilization.

Sites and Domain Controllers

A domain controller is automatically placed within a site during the server promotion process. DCPromo checks for the defined sites during the promotion process; if the server's IP address falls within the range of a defined subset, the server is automatically placed within the site associated with that subnet.

If no subnets are associated with site objects, the server is placed in the default site, which is named Default-First-Site. If the IP address of the server does not fall within a range that is defined, the server is placed in the Default-First-Site. Sites are automatically assigned only during the initial promotion; if a DC configuration or physical location changes significantly, the DC must also be moved to another site via the AD Sites and Services snap-in.

Multihomed servers can belong to only a single site. When a multihomed server is promoted, DCPromo selects the site at random from the ones that the server matches. If you do not agree with the selection, you can move the DC to another site via the AD Sites and Services snap-in.

Creating a Site

Sites are created via the AD Sites and Services snap-in, shown in Figure 2-13. Windows 2000 creates the first site automatically when Active Directory is installed. This site is named Default-First-Site, and it includes all the DCs. Additional sites must be created manually. To create a site, open the snap-in, and then open the context menu of the Sites folder. Select the New Site option to create a new site.

The New Object–Site screen, shown in Figure 2-14, allows you to enter the name of the remote site and to select the site link for the site. Windows 2000 creates a default site link called DEFAULTIPSITELINK that you can use to establish the replication process of the Active Directory service. This default site link uses remote procedure calls (RPC) over TCP/IP and will use any available route to the remote site for replication. If explicit site links have been previously defined, those site links will show up in the lower portion of the New Object–Site screen.

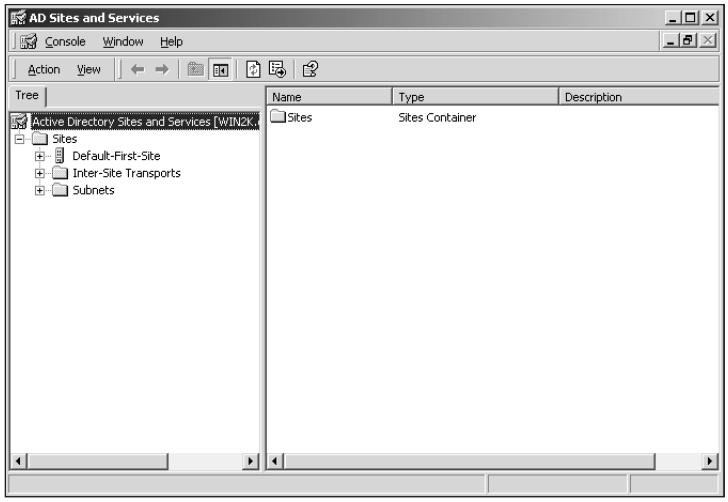

Figure 2-13 AD Sites and Services snap-in

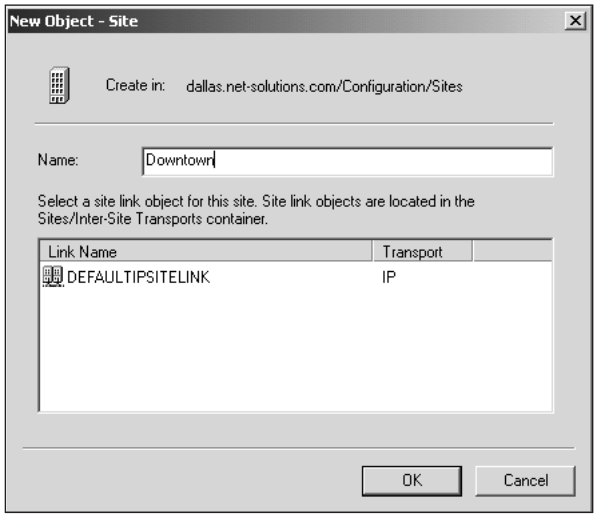

Figure 2-14 Creating a new site

Once the site is defined, you must undertake several other steps before the site can be activated within the Active Directory structure. These steps are nicely delineated in the dialog box that follows the creation of a new site, as shown in Figure 2-15. To finish configuring a site, you must do the following:

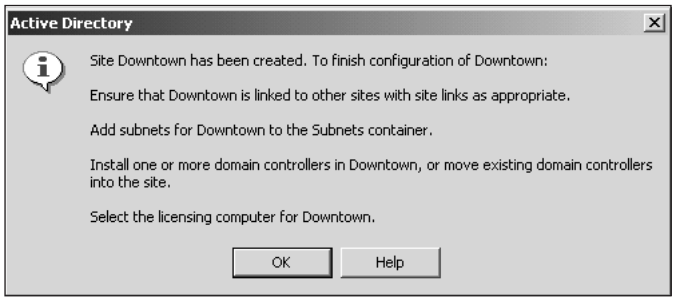

Figure 2-15 Required configuration steps for a new site

- Add appropriate IP subnets to the site.
- Install or move a DC or DCs into the site. Although a DC is not required for a site, it is strongly recommended.
- Connect the site to other sites with the appropriate site link.
- Select a server to control and monitor licensing within the site.

Once you've completed these steps, the site is then added to the Active Directory structure and the replication is automatically configured by Windows 2000.

Site Links

A site is a subnet or selection of subnets that are connected via a high-speed connection. The sites themselves are connected via site links. **Site links** are low bandwidth or unreliable/occasional connections between sites. In general, any connection between locations slower than LAN speeds is considered a site link. WAN links (such as frame relay connections) are examples of site links, as are high-speed links that are saturated and have a low effective bandwidth.

Site links are not automatically generated by Windows 2000. Instead, you create the site links through the AD Sites and Services snap-in. The site links are the core of Active Directory replication. The links can be adjusted for replication availability, bandwidth costs, and replication frequency. Windows 2000 uses this information to generate the replication topology for the sites, including the schedule for replication.

Windows 2000 DCs represent the inbound replication through a special object known as a **connection object**. Active Directory uses site links as indicators for where it should create connection objects, and connection objects use the physical network connections to replicate directory information. Each DC creates its own connection objects for replication within a site (**intrasite replication**). For replication between sites (**intersite replication**), one DC within each site is responsible for evaluating the replication topology. The DC creates the connection objects appropriate to that topology. The server that is responsible for evaluating and creating the topology for intersite replication is known as the Inter-Site Topology Generator (ISTG).

Site links, like trusts, are transitive. This means that DCs in one site can replicate with DCs in any other site within the enterprise through these transitive links. In addition, explicit links can be created to enable specific replication paths between sites.

Creating a Site Link

Windows 2000 creates a default site link named, naturally enough, DEFAULTIP-SITELINK. This site link can be used to connect sites in simple network environments, but in more complicated enterprise environments, you should establish explicit site links.

To create a site link, first open the AD Sites and Services snap-in, as indicated in Figure 2-16. Open the Inter-Site Transports folder and then right-click on the appropriate transport protocol. Select New Site Link from the context menu to form a new link.

In our example, the name of the new site link is Uptown Frame Link. Although the name of the link is arbitrary, good administrative practice dictates the name should be something that identifies the link, the connected sites, and the type of link. Of course, the link could be named Bob, but that name would tend to confuse successors and co-workers.

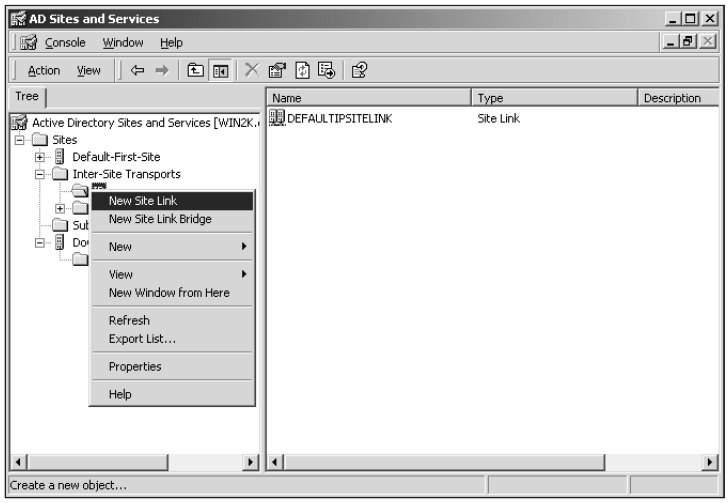

Figure 2-16 Creating a new site link

The next step is to select the linked sites from the left column in the New Object–Site Link dialog box. Click on Add to associate them with the link, as shown in Figure 2-17. A link must contain at least two sites; in general, a link will connect only two sites. If multiple sites exist at one physical location or are connected via a particular network path, however, then those sites could all share a single site link.

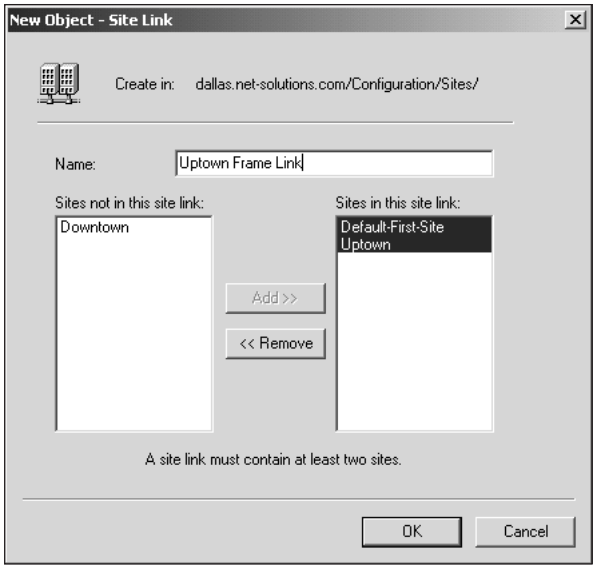

Figure 2-17 Naming the site link and associating the sites

Each site link has four properties that are important, as well as an optional descriptor. The properties are:

- *Name:* A name that uniquely identifies the site link. As discussed earlier, this name should clearly indicate the sites being linked and the speed/type of circuit.
- *Cost:* The relative speed of the link in relation to the other links within the topology. The cost has nothing to do with the actual monetary cost of the bandwidth. Links with a lower cost are faster, whereas links with higher costs are slower. The cost defaults to 100 on a new circuit.
- *Transport:* Indicates the type of transport used to replicate the directory information between the DCs. There are two options: synchronous RPC over a routed TCP/IP connection, and an asynchronous Simple Mail Transfer Protocol (SMTP) connection over the underlying mail transport network. This property is not set within the link properties, but is instead determined when the site link is first created.
- *Schedule:* Determines when the directory information is replicated between sites. This is determined by two elements: the replication frequency and the available times. The replication frequency is adjusted within the properties of the site link, as shown in Figure 2-18. The schedule is a listing of times that the site link is available to pass replication data. This schedule is adjusted through the Change Schedule option within the site link properties.

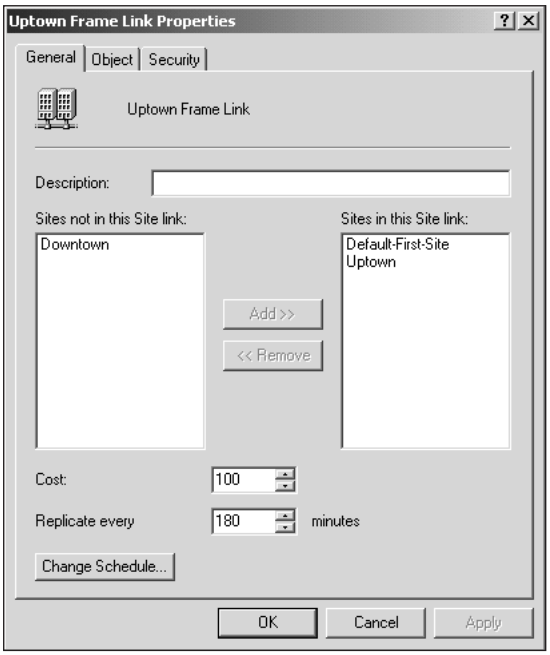

Figure 2-18 Site link properties

Bridgehead Servers

The replication topology between and among sites is generated automatically by Windows 2000. This topology is generated via a service known as the Knowledge Consistency Checker (KCC). The KCC service tries to establish at least two inbound replication connections to every DC; therefore, if a server becomes unavailable or uncommunicative, replication can still occur.

Within a site, all DCs are treated equally, but replication between sites is another matter. Windows 2000 prefers to funnel intersite replication to only a single DC. These preferred servers are known as **bridgehead servers**. The replicated data is first sent to the bridgehead server of a site, and is then replicated from that bridgehead server to the other DCs within the site.

If the preferred bridgehead server is unavailable, Active Directory will use an alternate replication path.

Chapter Summary

- Active Directory brings a whole new element to Windows domain functionality, and also brings with it new terminology and concepts. The core concept of Active Directory is that everything within the network environment is defined as an **object**. The directory is simply that—a database that stores the object, information about the object, and that object's relationship to the other objects. The types of objects that are allowed in the directory are defined by the schema. Windows 2000 automatically installs a default schema, but you can make modifications to those default definitions if desired.
- The basic unit of Windows networking through the years has been the **domain**. A domain is simply a collection of computers, users, and other objects that share a common security boundary and policy. The traditional Windows NT domain is a standalone entity. If users require access to resources within another domain, a special connection must be created between the domains. These special connections are known as **trusts**. Windows 2000 treats domains somewhat differently: An underlying relationship exists between Windows 2000 domains that share a common schema.
- Windows 2000 domains that share a common schema, namespace, and Global Catalog share **transitive two-way trusts**. These transitive trusts form a structure known as a **domain tree**. If a combination of domains share a Global Catalog and common schema, but occupy a different namespace, then the structure is actually a collection of domain trees. This structure is known as a **domain forest**.
- Each domain requires at least one domain controller. The DCs within a domain contain the Active Directory database and use that database to authenticate users, services, and computers. DCs record changes to the database and replicate these changes to the other DCs. Every DC has the ability to pass changes to other DCs. This process is known as **multi-master replication**.
- One of the design requirements for Windows 2000 was the ability to log in throughout the domain tree. Obviously, this requirement means that the DCs for one domain need to know which users, computers, and other objects are defined in the other domains. However, replication of all directory information in all domains throughout a domain tree or forest would be cumbersome and very bandwidth intensive. Windows 2000 resolves this issue through the use of Global Catalog Servers. Global Catalog Servers are DCs that contain a subset of information from other domains in a domain forest. The subset contains the most commonly used objects and greatly speeds both searches within the domain forest and logon services.
- Despite the use of multi-master replication, several functions of Active Directory replication need to be controlled from a single point. These functions are handled by specially designated DCs known as **operations masters**. Operations masters handle five functions: the schema master, domain naming master, relative ID master, primary DC emulator, and infrastructure master. Windows 2000 assigns each of

these roles to the first DC installed in a domain forest, but the roles can be adjusted as desired by the administrator.

- Windows 2000 handles replication over a WAN very differently from earlier versions of Windows NT. Windows 2000 groups well-connected computers into sites. A **site** is an area of high-speed connectivity. Sites are linked by slower links, usually WAN links. Sites are associated with a particular subnet or subnets, and usually include a DC. Replication within a site is optimized to prevent latency; replication between sites is optimized to conserve bandwidth. The timing of the replication can be adjusted to take advantage of periods of lower network utilization.
- Replication topology between and among sites is controlled by the Knowledge Consistency Checker (KCC). The KCC attempts to connect each DC to two other DCs for replication purposes. The KCC runs every 15 minutes to verify connectivity and to adjust the replication structure as needed. Intersite replication is handled via preferred DCs that replicate between sites and then pass the replicated data to the DCs within the sites.
- One important thing to recall is that no direct relationship exists between sites and domains. A site can encompass multiple domains, and a domain can encompass multiple sites.